

《软件工程》

图书基本信息

《软件工程》

内容概要

《软件工程》主要以工程化的软件开发方法为主导，系统、全面地介绍这门课程的原理、方法及应用。既注重系统性和科学性，又注重实用性；既有原理性论述，又有丰富的实例与之配合。全书共分9章，内容包括软件工程概论、软件计划与可行性研究、软件需求分析、软件结构设计、软件详细设计、软件编码设计、软件测试、软件维护和面向对象的软件开发技术概述。

《软件工程》可作为高等院校“软件工程”课程的教材或教学参考书，也可作为从事计算机工作的科技人员学习软件工程的参考书。

第1章 软件工程概论1.1 软件综述1.1.1 软件的发展1.1.2 软件的定义1.1.3 软件的特性1.1.4 软件分类1.2 软件危机1.3 软件工程1.3.1 软件工程的定义1.3.2 软件工程的基本原理1.3.3 软件工程研究的内容1.3.4 软件工程方法学1.4 软件生命周期1.5 软件开发模型1.5.1 瀑布模型1.5.2 快速原型模型1.5.3 增量模型1.5.4 螺旋模型1.5.5 基于构件的开发模型1.5.6 智能模型1.6 小结习题1第2章 软件计划与可行性研究2.1 软件计划2.1.1 软件的作用范围2.1.2 资源需求2.1.3 进度安排2.2 可行性研究2.2.1 可行性研究的任务2.2.2 可行性研究的步骤2.2.3 可行性研究的CASE工具2.3 成本 / 效益分析2.3.1 软件开发成本估计2.3.2 效益度量方法2.3.3 效益分析方法2.4 可行性研究报告的书写2.5 小结习题2第3章 软件需求分析3.1 需求分析的任务3.2 需求分析的步骤3.3 需求分析方法和原则3.4 需求分析的CASE工具3.4.1 层次方框图3.4.2 Warnier图3.4.3 IPO图3.4.4 E-R图3.4.5 状态迁移图3.4.6 Petri网3.5 软件需求验证3.5.1 软件需求规格说明的主要内容3.5.2 软件需求的验证3.6 小结习题3第4章 软件结构设计4.1 概述4.1.1 软件设计的过程4.1.2 软件结构设计的目标4.1.3 软件结构设计的任务4.1.4 软件结构设计的方法4.2 结构化设计的概念和原理4.2.1 抽象和细化4.2.2 自顶向下, 逐步求精4.2.3 信息隐藏和局部化4.2.4 模块化4.2.5 模块独立性4.3 结构设计的原则4.4 结构设计的图形工具4.4.1 层次图和HIPO图4.4.2 结构图4.5 面向数据流的设计方法4.5.1 数据流图的类型4.5.2 结构设计过程4.5.3 变换分析4.5.4 事务分析4.5.5 软件结构的优化4.6 小结习题4第5章 软件详细设计5.1 结构化程序设计5.1.1 结构化的控制结构5.1.2 逐步细化的实现方法5.1.3 结构化程序设计的特点5.2 详细设计的工具5.2.1 程序流程图5.2.2 盒图 (N-S) 图5.2.3 问题分析图 (PAD图) 5.2.4 判定表5.2.5 判定树5.2.6 过程设计语言 (PDL) 5.3 面向数据流的设计方法5.3.1 Jackson图5.3.2 Jackson方法5.4 程序复杂度的概念及度量方法5.4.1 程序图5.4.2 程序复杂度的度量方法5.5 小结习题5第6章 软件编码设计6.1 编码设计的目的6.2 编码的工具语言6.2.1 程序设计语言的分类6.2.2 程序设计语言的特性6.2.3 程序设计语言的选择6.3 编码风格及软件效率6.3.1 编码风格6.3.2 软件效率6.4 小结习题6第7章 软件测试7.1 软件测试概述7.1.1 软件测试的概念7.1.2 软件测试的目的7.1.3 软件测试的原则7.1.4 软件测试方法7.2 软件测试过程模型7.3 软件开发过程的测试步骤7.3.1 单元测试7.3.2 集成测试7.3.3 系统测试7.3.4 验收测试7.4 软件测试用例的设计7.4.1 测试用例概述7.4.2 白盒测试法的用例设计7.4.3 黑盒测试法的用例设计7.5 软件调试7.5.1 调试原则7.5.2 软件调试的步骤7.5.3 软件调试的策略7.6 小结习题7第8章 软件维护8.1 软件维护的任务和分类8.1.1 改正性维护 (correctivemaintenance) 8.1.2 适应性维护 (adaptivemaintenance) 8.1.3 完善性维护 (perfectivemaintenance) 8.1.4 预防性维护 (preventivemaintenance) 8.2 软件维护的特点8.2.1 非结构化维护和结构化维护8.2.2 软件维护的困难性8.2.3 软件维护的费用8.3 软件的可维护性8.3.1 软件可维护性的定义8.3.2 影响软件可维护性的因素8.3.3 文档8.3.4 软件可维护性的定量度量8.3.5 提高软件可维护性的方法8.4 软件维护活动8.4.1 软件维护组织机构8.4.2 软件维护申请报告8.4.3 软件维护工作流程8.4.4 维护档案记录8.4.5 维护评价8.5 小结习题8第9章 面向对象的软件开发技术概述9.1 面向对象的概念9.2 面向对象的建模9.2.1 面向对象方法的开发模型9.2.2 面向对象方法9.2.3 面向对象模型9.3 面向对象的分析9.3.1 面向对象分析的3个模型与5个层次9.3.2 构造对象模型9.3.3 构造对象动态模型9.3.4 建立功能模型9.3.5 定义服务9.4 面向对象的设计9.4.1 面向对象设计的概念9.4.2 问题域子系统设计9.4.3 人机交互子系统设计9.4.4 任务管理子系统设计9.4.5 数据管理子系统设计9.4.6 服务与关联的设计9.4.7 面向对象设计的优化9.5 面向对象的实现9.5.1 面向对象实现的技术支持9.5.2 面向对象语言选择9.5.3 程序设计风格9.5.4 面向对象的测试9.6 组件技术简介9.6.1 组件的概念及特点9.6.2 组件模型9.6.3 组件开发模式9.7 小结习题9参考文献

版权页：插图：（2）硬件资源硬件资源包括开发系统、运行平台和新系统的其他硬件。对于一个专业的开发机构来说，为了提高软件的开发效率，常常配备有专门的软件开发系统（包含硬件和软件）。这类开发系统中提供了大量的开发工具，这些开发工具不仅支持软件开发的功能，还支持硬件开发和系统仿真等。在使用开发系统进行开发时，软件计划中主要考虑的是开发系统允许使用的时间安排和机时的消耗问题，这都涉及成本的估算和进度安排。在开发某些专用软件时，还需要考虑某些特殊的硬件资源。例如，在开发自动排版软件时，到达某一时刻就需要一台照排机等。因此，在制订计划时，需要全面细致的考虑。（3）软件资源软件资源包括支撑软件和可重用软件两大类。支撑软件的范围极为广泛，它包含了所有的工具软件。最一般的支撑软件是操作系统、编译程序、编辑连接程序等。另外，在软件工程的各个阶段都有相应的支撑软件可供采用。在需求分析阶段，有自动的规格需求说明制作程序；在设计阶段，有设计语言处理程序、流程图、框图生成程序；在测试阶段，有测试驱动程序和分析程序等。适当地使用支撑软件，可以提高软件开发速度和软件生产率，也可以提高软件质量。但为了要使支撑软件能在开发系统上运行，可能要花费巨大的人力资源和高昂的资金成本。故制订计划者在选择支撑软件时，要充分考虑到效益/成本问题。如果没有对可重用性的认识，任何关于软件资源的讨论都将是不完整的，可重用性是指软件建筑块的创建及重用。Bennatan建议在计划进行过程中应该考虑的可重用软件资源：

- 可直接使用的构件：已有的，能够从第三方厂商获得或已经在以前的项目中开发过的软件。这些构件已经经过验证及确认且可以直接用在当前的项目中。
- 具有完全经验的构件：已有的为以前类似于当前要开发的项目建立的规约、设计、代码或测试数据。当前软件项目组的成员在这些构件所代表的应用领域中具有丰富的经验。因此，对于这类构件进行所需的修改其风险相对较小。

《软件工程》

编辑推荐

《软件工程》：21世纪高等学校本科系列教材·计算机科学与技术专业

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com